Rule-Enhanced Graph Learning

Ali Khazraee[1] Abdolreza Mirzaei [1 2] Majid Farhadi [2 3] Parmis Naddaf [2] Kiarash Zahirnia [2] Mohammad Salameh [4] Kevin Cannons [4] Richard Mar [2] Mingyi Wu [2] Oliver Schulte [2]

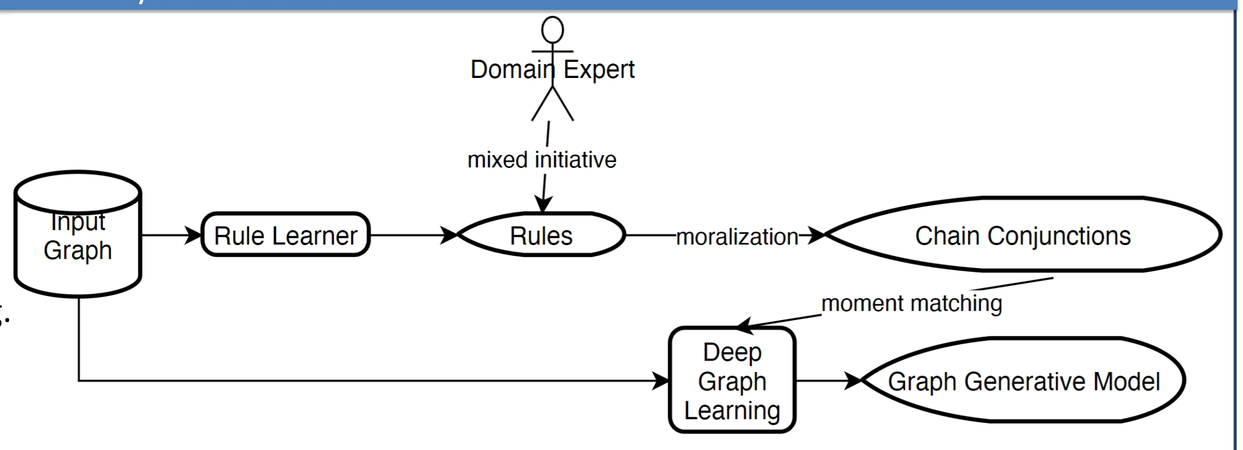
[1]Isfahan University of Technology, [2]Simon Fraser University, [2]University of Alberta, [4]Huawei Technologies Ltd.

Overview

- **Enhancing deep generative models with first-order logic rules.**
- A kind of *neuro-symbolic AI.*
- Key idea is rule moment matching
- Algorithmic contribution: novel matrix multiplication method for computing the expected rule instance count of a Variational Graph Autoencoder (VGAE).

System Overview

- **System Components:** Integrates SRL with deep GGMs.
- **Requirement**: The expected rule instance count from the model should match the observed count in the training graph.
- **GGM Training objective**: Maximize the GGM likelihood while ensuring rule moment matching.
- **Example:** Calculate the number of triangles in a graph using the third power of its adjacency matrix, A^3.

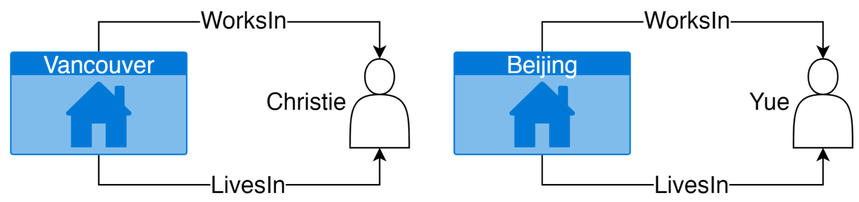


Contributions

- A **new objective function** for enhancing generative graph learning with domain knowledge represented by logical rules.
- A new **matrix multiplication algorithm for counting the number of rule instances** in a graph.
- A new **algorithm for estimating the expected number of rule instances** for a Variational Graph Autoencoder model, based on matrix multiplication.
- Two extensions of the previous VGAE model:
  - **VGAE+** generates node features and edge types.
  - **VGAE+R** train a VGAE+ model that matches rule instance counts.

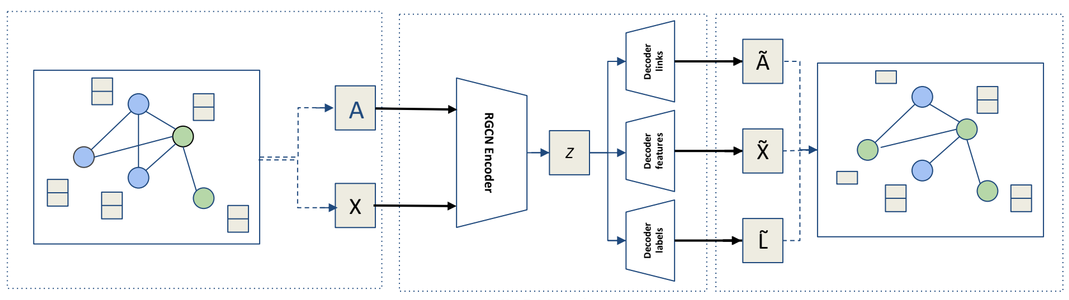
First Order Logic Rules

- Different enhancement methodologies are appropriate for different types of knowledge (e.g., knowledge from models, humans, external sources).
- We leverage knowledge from First Order Logic rules.
- An example rule would be "if person X works in city Y, then X lives in city Y (with probability p)



VGAE+

- The VGAE+ model extends VGAE by adding decoders for node features and labels.
  - comprehensive generative model for attributed labeled graphs
  - enabling the independent generation of links, features, and labels.
- Decoders:
  - Adjacency matrix decoder: Generates links based on node embeddings. (eq.1)
  - Feature decoder: Generates node features from node embeddings. (eq.2)
  - Label decoder: Generates node labels from node embeddings. (eq.3)

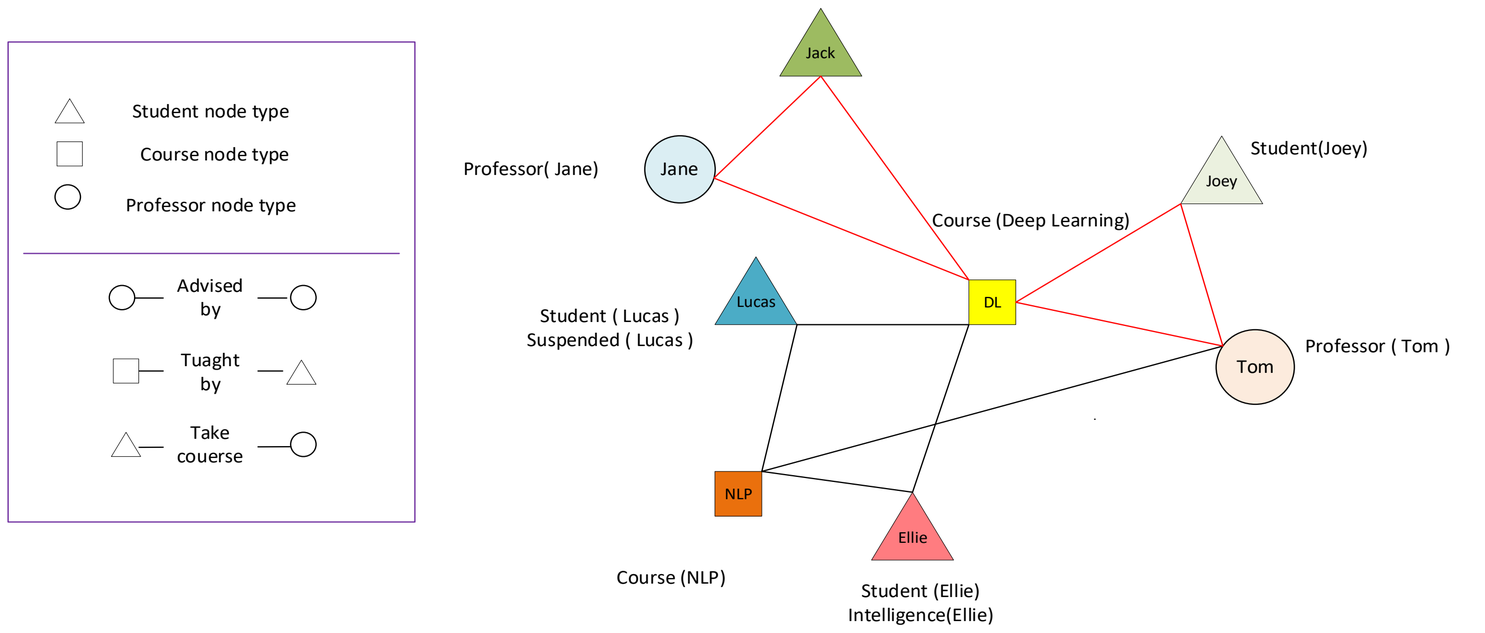


$$eq1.\ p_\theta(A|z) = \prod_{r=1}^{T} \prod_{u,v} p_\theta(A_r[u,v]|z[u],z[v]) \quad eq2.\ p_\psi(X|z) = \prod_u p_\psi(X[u]|z[u]) \quad eq3.\ p_\phi(L|z) = \prod_u p_\phi(L[u]|z[u])$$

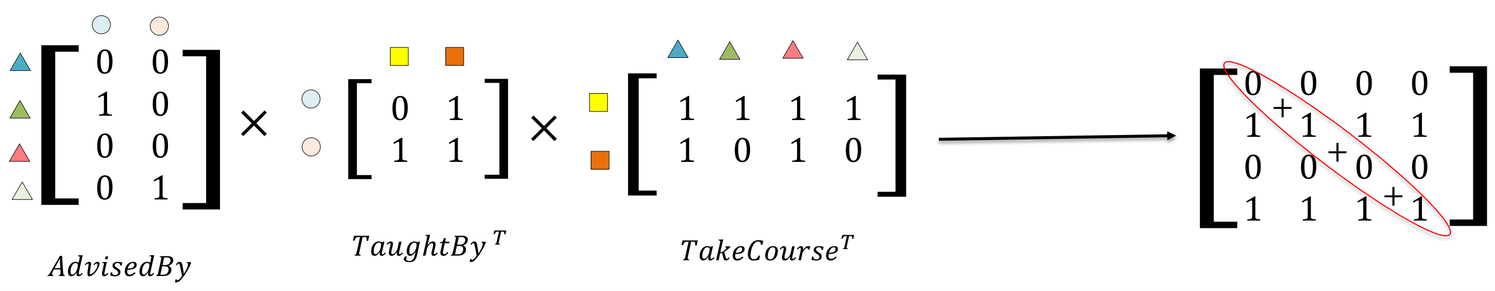
Matrix multiplication method / VGAE+R

- VGAE+R :

$$\mathcal{L}(\theta,\psi,\phi) = \mathrm{KL}(q_\phi(z|X,A)||p(z) - E_{z\sim q}(z|\mathrm{X},\mathrm{A}) + [\alpha\times\ln p_\theta(\mathrm{A}|\mathrm{z}) + \beta\times\ln p_\psi + \gamma\times\ln p_\phi(L|z) + {}^{\lambda}/_{k}\sum_{i=1}^{k}\rho(n_i(D), E[n_i(G|z)]$$



- Motif: A motif in a graph, shown as a red line, is a repeated subgraph pattern represented by the rule.
- Rule: $AdvisedBy(student, professor), TaughtBy(course, professor), TakeCourse(student, course)$



Results – Node Classification

Dataset	Metric	Model	
		VGAE+	VGAE+R
Cora	Precision	0.7952	**0.8556**
	F1	0.7178	**0.8440**
	Recall	0.8284	**0.8542**
Citeseer	Precision	0.7265	**0.7579**
	F1	0.6958	**0.7251**
	Recall	0.7233	**0.7639**
IMDB	Precision	0.6328	0.6337
	F1	**0.6282**	0.6219
	Recall	**0.6297**	0.6250
ACM	Precision	**0.9675**	0.9323
	F1	**0.8623**	0.7925
	Recall	**0.9624**	0.9396

Results – Graph Realism

Dataset	Cora	Citeseer	IMDB	ACM
VGAE+	2.8E+12	6.8E+11	1.7E+19	**1.9E+15**
VGAE+R	**1.2E+10**	**2.6E+9**	**7.8E+15**	4.5E+15

Results – Count Distance

Dataset	Cora	Citeseer	IMDB	ACM
VGAE+	4233.6	4085.6	137612.3	36572.1
VGAE+R	**797.2**	**523.56**	**14373.3**	**12984.5**

Propositions

- The expected conjunction count given a set of node embeddings can be computed as the conjunction count in the expected graph. (eq1)

$$eq1.\ E[n_\phi(G)|z] = n_\phi(\widetilde{G_Z})$$

- The (u,v)-th entry of O_k counts the number of groundings of a centered chain conjunction ϕ of length k in a graph G where $U_1 = u$ and $V_p = v$. Therefore

$$eq2.\ n_\phi(G) = \sum(O_k(\phi))$$

Results – Link Prediction

Dataset	Model type	ACM		Cora	Citeseer	IMDB	
		Author - Paper	Paper - Subject	Paper - Paper	Paper - Paper	Actor - Movie	Director - Movie
AUC	VGAE+	0.974823	0.974056	0.920903	0.906741	0.903155	0.886872
	VGAE+R	0.976939	0.974062	**0.932803**	0.880990	0.9074	0.890821
AP	VGAE+	0.994472	0.996084	0.90638	0.906388	0.920079	0.894908
	VGAE+R	0.996288	0.997294	**0.935509**	0.915649	0.925500	0.894994